



FINITE ELEMENT ANALYSIS OF SPACE TRUSS USING MATLAB

P. Sangeetha¹, P. Naveen Kumar¹ and R.Senthil²

¹SSN College of Engineering, Chennai, India

²College of Engineering, Guindy, Anna University, Chennai, India

ABSTRACT

One of the interesting engineering application is space truss, a three dimensional element, particularly used as roof for industrial and commercial structure spanning large distances. Analysis of space truss can be performed by many commercial FEA software available in the market. This paper concerns the current growth of MATLAB based program which analyze the space truss step by step as done in Finite Element Analysis. Using MATLAB all stages of calculations have been done to solve the space truss and verify with that of published experimental data.

Keywords: space truss, MATLAB, finite element analysis.

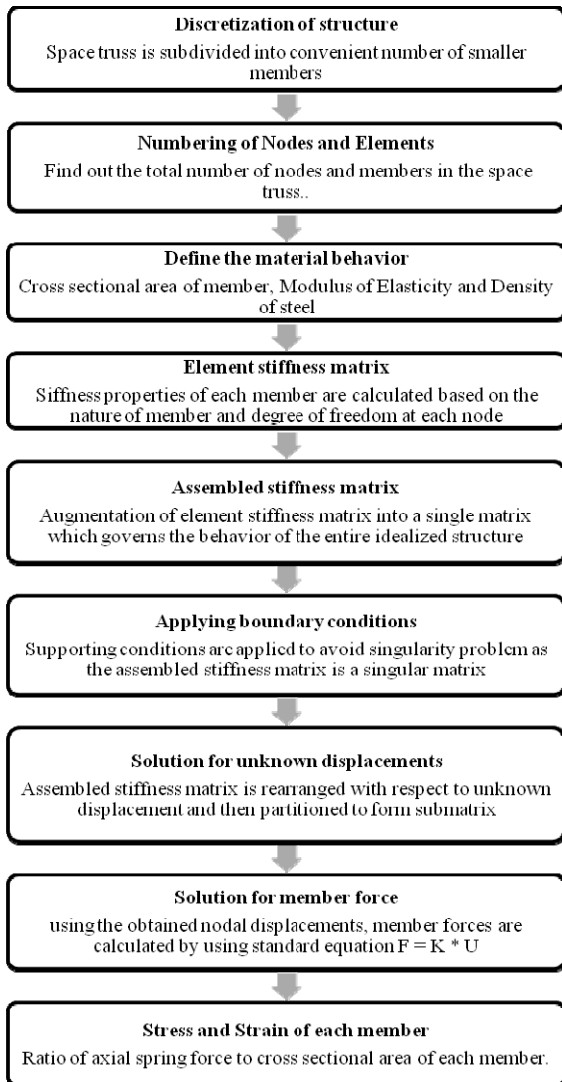
INTRODUCTION

Space Truss is a lightweight rigid structure consists of members and nodes interlocking in a triangular geometric pattern. The inherent rigidity of triangular geometric pattern derives its strength. On the application of imposed load, tension and compression loads are transmitted along the length of the member. The advantage of using space trusses as roof structure is to provide rigidity along all the three directions comparatively higher strength than the normal truss. Finite Element Analysis of a space truss involves many matrix operations. Sometimes it is tedious to solve manually if the size of the matrix goes higher. Matrix is the fundamental object of MATLAB. MATLAB is a trademark of The Math Works, Inc., USA and majorly designed to perform matrix operations. MATLAB based program is simple coding system with matrix functions, conditionals (if and switch), loops (for and while) and Graphics (2Dplots and 3Dplots). MATLAB is a simpler package but its application stands in all fields of science and Engineering. Schmidt et al, have studied that with greater degrees of freedom, space trusses became more sensitive to compression members, joints and stiffness of the member-node joints normally neglected in the design and it consequently proven to be failure of the structure. A.I. EI-Sheikh, introduced simple jointing system provides ductility and enhances the ability to redistribute forces that the reduction of overall component cost. Ahmed Farhan Kadhum analysed the space trusses by using ANSYS software and found that ANSYS provide graphical option to simulate the real behavior of the structure under the application of load and a good agreement between the analytical and experimental load-deflection values. O. Cagalayan and E. Yuksel made a study on the possible reasons for the collapse of a roof structure using SAP. He concluded that the most probable reason behind the collapse were improper substitution of snow load intensity and some mistakes in the elastic design of the truss. Per-Ol of Persson and Gilbert Strang have created a mesh with a simple MATLAB code to find mesh points and triangulation. They concluded that the algorithm can be useful in other areas and the distance

function representation was effective for moving boundary problems

FINITE ELEMENT FORMULATION

Finite Element Analysis is a numerical method to solve Engineering problems and Mathematical physics. A space truss is subdivided into smaller elements called members. Then the assemblage of these members connected at a finite number of joints called Nodes. The properties of each type of member is obtained and assembled together and solved as whole to get solution. Static analysis of space truss is done by stiffness method where formulation is simpler for most structural analysis problems. Algorithms for solving space truss problem are formulated below:



SPACE TRUSS ELEMENT

The space truss element is a three-dimensional finite element with both local and global coordinates. It is characterized by linear shape functions. It has modulus of elasticity E, cross sectional area A, and length L. Each

space truss element has two nodes and is inclined with angles θ_x , θ_y , and θ_z measured from the global X, Y, and Z axes respectively, to the local x axis as shown in figure. Let $C_x = \cos\theta_x$, $C_y = \cos\theta_y$, and $C_z = \cos\theta_z$. Figure-1 shows the space truss coordinates and its element stiffness matrix.

MATLAB CODES FOR SPACE TRUSS

This function returns the element length gives the coordinates of their respective nodes and the element stiffness matrix for space truss element with m

odulus of elasticity E, cross-sectional area A, Length L, and angles thetax, thetay, and theaz. It returns the 6x6 element stiffness matrix k.

```

function y = SpaceTrussElementLength(x1,y1,z1,x2,y2,z2)
%SpaceTrussElementLength This function returns the length of the space truss element whose first node has coordinates (x1,y1,z1) and second node has coordinates (x2,y2,z2).
y = sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1) + (z2-z1)*(z2-z1));
    
```

function y = Space Truss Element Stiffness (E,A,L,thetax,thetay,thetaz)

%Space Truss Element Stiffness This function returns the element stiffness matrix for a space truss element with modulus of elasticity E, cross-sectional area A, length L, and angles the tax, the tay, the taz (in degrees). The size of the element stiffness matrix is 6 x 6.

```

x = thetax*pi/180;
y = thetay*pi/180;
z = thetaz*pi/180;
Cx = cos(x);
Cy = cos(y);
Cz = cos(z);
w = [Cx*Cx Cx*Cy Cx*Cz; Cy*Cx Cy*Cy Cy*Cz; Cz*Cx Cz*Cy Cz*Cz];
y = E*A/L*[w -w; -w w];
    
```

$$C_x = \frac{x_2 - x_1}{L}; C_y = \frac{y_2 - y_1}{L}; C_z = \frac{z_2 - z_1}{L}$$

$$k = \frac{EA}{L} \begin{bmatrix} C_x^2 & C_x C_y & C_x C_z & -C_x^2 & -C_x C_y & -C_x C_z \\ C_y C_x & C_y^2 & C_y C_z & -C_y C_x & -C_y^2 & -C_y C_z \\ C_z C_x & C_z C_y & C_z^2 & -C_z C_x & -C_z C_y & -C_z^2 \\ -C_x^2 & -C_x C_y & -C_x C_z & C_x^2 & C_x C_y & C_x C_z \\ -C_y C_x & -C_y^2 & -C_y C_z & C_y C_x & C_y^2 & C_y C_z \\ -C_z C_x & -C_z C_y & -C_z^2 & C_z C_x & C_z C_y & C_z^2 \end{bmatrix}$$

Figure-1. Space truss coordinates and its stiffness matrix.



www.arpnjournals.com

%Space Truss Element Assemble This function assembles the element stiffness matrix k of the space truss element with nodes i and j into the global stiffness matrix K. This function returns the global stiffness matrix K after the element stiffness matrix k is assembled. **% Stiffness Matrix Construction**

```
k = zeros(3*G,3*G);
for i=1:E
for j=1:3
for h=1:3

k(3*(11(i)-1)+h,3*(11(i)-1)+j) = k(3*(11(i)-1)+h,3*(11(i)-1)+j) + k_elem(h,j,i);
k(3*(11(i)-1)+h,3*(12(i)-1)+j) = k(3*(11(i)-1)+h,3*(12(i)-1)+j) + k_elem(h,j,i);
k(3*(12(i)-1)+h,3*(11(i)-1)+j) = k(3*(12(i)-1)+h,3*(11(i)-1)+j) + k_elem(h,j,i);
k(3*(12(i)-1)+h,3*(12(i)-1)+j) = k(3*(12(i)-1)+h,3*(12(i)-1)+j) + k_elem(h,j,i);
```

%Assemble Matrix Construction

```
K = zeros(3*G,3*G);
for i=1:G
for j=1:E
if(11(j)==i || 12(j)==i)
K(3*i-2,3*i-2) = (Ro(j)*Area(j)*L(j))/2 + K(3*i-2,3*i-2);
K(3*i-1,3*i-1) = (Ro(j)*Area(j)*L(j))/2 + K(3*i-1,3*i-1);
K(3*i,3*i) = (Ro(j)*Area(j)*L(j))/2 + K(3*i,3*i);
```

%Space Truss Element Assemble This function assembles the element stiffness matrix k of the space truss element with nodes i and j into the global stiffness matrix K. This function returns the global stiffness matrix K after the element stiffness matrix k is assembled. **%Stiffness Matrix Construction**

```
k = zeros(3*G,3*G);
for i=1:E
for j=1:3
for h=1:3
k(3*(11(i)-1)+h,3*(11(i)-1)+j) = k(3*(11(i)-1)+h,3*(11(i)-1)+j) + k_elem(h,j,i);
k(3*(11(i)-1)+h,3*(12(i)-1)+j) = k(3*(11(i)-1)+h,3*(12(i)-1)+j) + k_elem(h,j,i);
k(3*(12(i)-1)+h,3*(11(i)-1)+j) = k(3*(12(i)-1)+h,3*(11(i)-1)+j) + k_elem(h,j,i);
k(3*(12(i)-1)+h,3*(12(i)-1)+j) = k(3*(12(i)-1)+h,3*(12(i)-1)+j) + k_elem(h,j,i);
%Assemble Matrix Construction
K = zeros(3*G,3*G);
for i=1:G
for j=1:E
if(11(j)==i || 12(j)==i)
K(3*i-2,3*i-2) = (Ro(j)*Area(j)*L(j))/2 + K(3*i-2,3*i-2);
K(3*i-1,3*i-1) = (Ro(j)*Area(j)*L(j))/2 + K(3*i-1,3*i-1);
K(3*i,3*i) = (Ro(j)*Area(j)*L(j))/2 + K(3*i,3*i);
```

Function y = Space Truss Element Force (E, A, L, the tax, the tay, the taz,u)

This function returns the element force given the modulus of elasticity E, the cross-sectional area A, the length L, the angles the tax, the tay, the taz (in degrees), and the element nodal displacement vector u.

```
x = thetax * pi/180;
w = thetay * pi/180;
v = thetaz * pi/180;
Cx = cos(x);
Cy = cos(w);
Cz = cos(v);
y = E*A/L*[-Cx -Cy -CzCx Cy Cz]*u;
```

Function y = Space Truss Element Stress (E,L, the tax, the tay, the taz,u)

This function returns the element stress given the modulus of elasticity E, the length L, the angles the tax, the tay, the taz (in degrees), and the element nodal displacement vector u.

```
x = the tax * pi/180;
w = the tay * pi/180;
v = the taz * pi/180;
Cx = cos(x);
Cy = cos(w);
Cz = cos(v);
y = E/L*[-Cx -Cy -CzCx Cy Cz]*u;
%Plot Undeformed and Deformed shapes
for i=1:E
x_bef(1,i)=x(11(i));
x_bef(2,i)=x(12(i));
y_bef(1,i)=y(11(i));
y_bef(2,i)=y(12(i));
z_bef(1,i)=z(11(i));
z_bef(2,i)=z(12(i));
x_aft(1,i)=x(11(i))+Q(1+3*(11(i)-1),2);
x_aft(2,i)=x(12(i))+Q(1+3*(12(i)-1),2);
y_aft(1,i)=y(11(i))+Q(2+3*(11(i)-1),2);
y_aft(2,i)=y(12(i))+Q(2+3*(12(i)-1),2);
z_aft(1,i)=z(11(i))+Q(3+3*(11(i)-1),2);
z_aft(2,i)=z(12(i))+Q(3+3*(12(i)-1),2);
end
x_offset=.2*(max(x)-min(x));
y_offset=.2*(max(y)-min(y));
z_offset=.2*(max(z)-min(z));
xmin=min(x)-x_offset;
xmax=max(x)+x_offset;
ymin=min(y)-y_offset;
ymax=max(y)+y_offset;
zmin=min(z)-z_offset;
zmax=max(z)+z_offset;
plot3(x_bef, y_bef, z_bef, 'b', x_aft, y_aft, z_aft, 'g')
axisequal;
axis([xmin xmax ymin ymax zmin zmax]);
view([1,1,1]);
```



Example -EI-sheikh model

Verification was done in order to check the accuracy and validity of using MATLAB based program for solving static analysis of a structure. A space truss model has been analyzed and compared with the published results of double layer space truss of size 4m x 4m. The space truss have top chord , bottom chord members and diagonal members. Numbering of nodal points was done for the space truss model along with coordinates with respect to global axes X, Y, and Z respectively. It consists of 61 nodes and 200 elements. Figure-2 shows the layout of the Space truss (EI-Sheikh model).

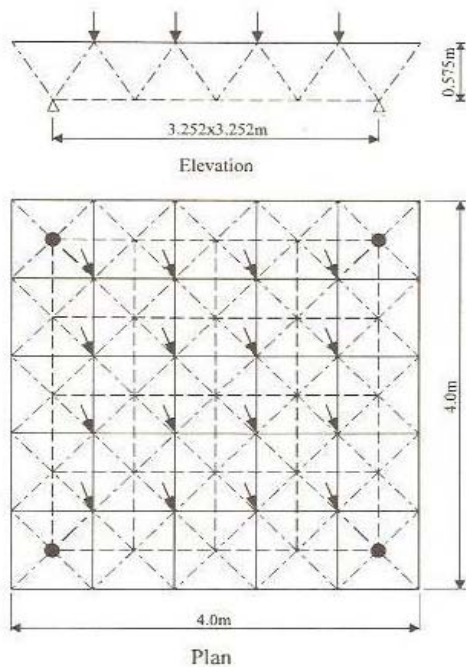


Figure-2. Layout of Space Truss (EI-Sheikh model).

MATERIALS

Circular steel tubes for bottom and diagonal chord members, and Rectangular steel tubes for top chord members were used. Young’s Modulus of material and density of steel was taken as $2 \times 10^5 \text{ N/mm}^2$ and 78.5 Kg/m^3 respectively. The dimension of the tubular members were shown in Table-1.

Table-1. Dimension of the Truss member.

S. No.	Specification of the members	
	Members	Size (mm)
1.	Top Chord Members	40 x 24 x 1.6
2.	Bottom Chord Members	28.58 x 6.3
3.	Diagonal Chord Members	28.58 x 6.3
4.	Corner Diagonal Chord	60.3 x 3.2

ELEMENT STIFFNESS MATRIX

The Element Stiffness matrix at each node was computed. Figure-3 shows the generation of the Element Stiffness matrix for the Element 1 which is connecting the node 1 and 2.

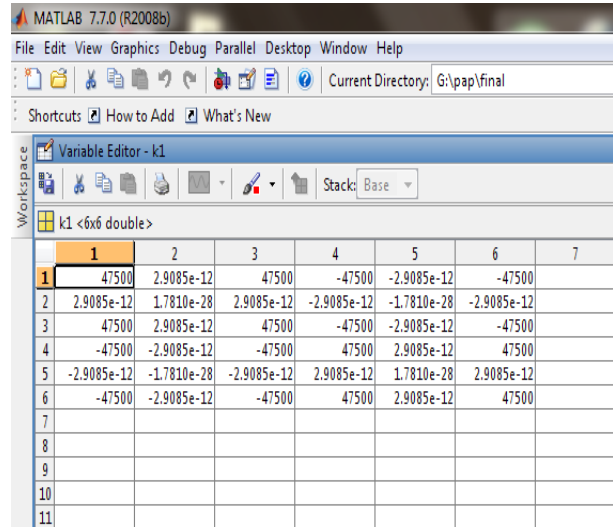


Figure-3. Element stiffness matrix for the Element 1.

SYSTEM STIFFNESS MATRIX

Element stiffness matrices were assembled to form a single matrix by making calls to the stiffness matrix of each element. Assembled Stiffness Matrix of given truss model having size $3(61) \times 3(61)$, where 61= total no. of nodes in a structure. The assembled stiffness matrix having a size of 183×183 was generated. A fixed support was provided at four end nodal points 58, 59, 60 and 61 and a concentrated load of 100KN was given at 16 nodal points of the top chord members. After obtaining the displacement at each node, form a nodal displacement vector U. Then the forces in the member was computed by using the standard formula $F = K * U$. Figure-4 shows deformed shape of the EI Sheikh Model using MATLAB.

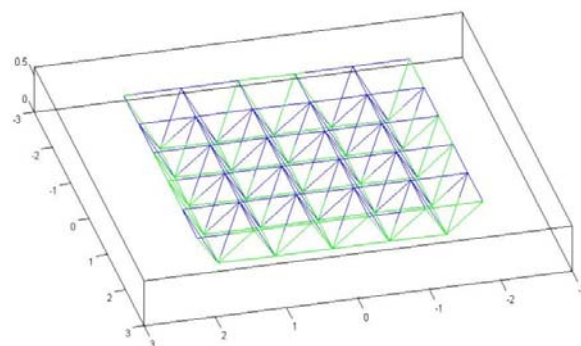


Figure-4. Deformed shapes using Mat lab – EI Sheikh Model.



RESULT AND DISCUSSIONS

The published experimental maximum deflection of the space truss of size 4m x 4m was about 5mm and from the Finite Element analysis using MATLAB the deflection was 4.4mm. The percentage decrease in deflection is about 12% from experimental result to MATLAB results. Figure-5 shows the load versus deflection curve between the experimental and MATLAB results.

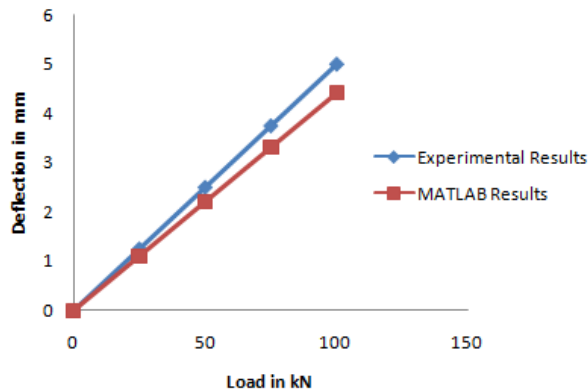


Figure-5. Load versus deflection curve for the.

CONCLUSIONS

1. The MATLAB coding was generated for the Double layer grid space truss structure.
2. The space truss of size 4m x 4m was analyzed using MATLAB against published experimental data. The maximum central deflection was about 12 % variation comparing with El-Sheikh model.

REFERENCES

- [1] Schmit L. C., Morgan P. R. and Hanaor A. 1982. Ultimate load testing of space trusses ‘ , Journal Of Structural Division , ASCE , Vol. 108, No. 7 , pp. 1324-1335.
- [2] El-sheikh A. and McConnel R.E. 1993. Experimental study of behavior of composite space trusses. Journal of Structural Engineering , Vol. 119, No. 3 , pp. 747-766.
- [3] El-sheikh A. and El-Barky H. 1996. Experimental Study of New space truss system. Journal of Structural Engineering , Vol. 122, No. 8 , pp. 845-853.
- [4] O.Caglaan, E.Yuksel. ‘Experimental and finite element investigations on the collapse of a Mero space truss roof structure –A case study.
- [5] Peter Kattan. ‘MATLAB Guide to Finite Elements – An interactive approach.
- [6] A. J. M. Ferrira. MATLAB codes for Finite Element Analysis.